

# Exception Handling Part - I

- Jayendra Khatod

## Objectives

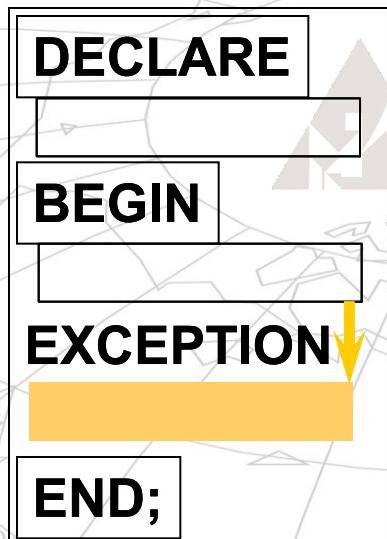
- **Define PL/SQL exceptions**
- **Recognize unhandled exceptions**
- **List and use different types of PL/SQL exception handlers**
- **Trap unanticipated errors**
- **Describe the effect of exception propagation in nested blocks**
- **Customize PL/SQL exception messages**

# Exception Handling with PL/SQL

- **What is an exception?**  
**Identifier in PL/SQL that is raised during execution**
- **How is it raised?**
  - An Oracle error occurs.
  - You raise it explicitly.
- **How do you handle it?**
  - Trap it with a handler.
  - Propagate it to the calling environment.

# Handling Exceptions

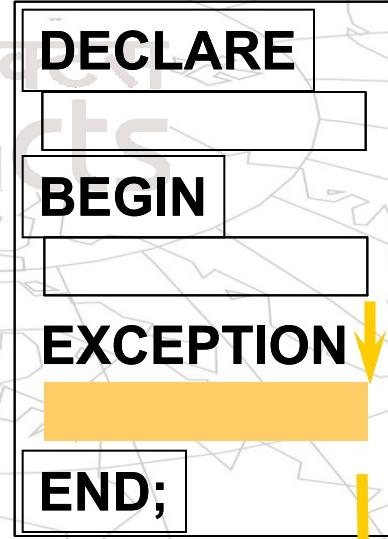
Trap the exception



Exception  
is raised

Exception  
is trapped

Propagate the exception



Exception  
is raised

Exception is  
not trapped

Exception propagates  
to calling environment

# Exception Types

- **Predefined Oracle Server**
  - **Non-predefined Oracle Server**
  - **User-defined**
- }
- Implicitly raised**
- Explicitly raised**

# Trapping Exceptions

- **Syntax – Exception Handler Section**

## EXCEPTION

```
WHEN exception1 [OR exception2 . . .] THEN
    statement1;
    statement2;
    .
    .
    .
[WHEN exception3 [OR exception4 . . .] THEN
    statement1;
    statement2;
    .
    .
    .
]
[WHEN OTHERS THEN
    statement1;
    statement2;
    .
    .
    .
]
```

# Trapping Exceptions Guidelines

- **WHEN OTHERS** is the last clause.
- **EXCEPTION** keyword starts exception-handling section.
- Several exception handlers are allowed.
- Only one handler is processed before leaving the block.

# Trapping Predefined Oracle Server Errors

- Reference the standard name in the exception-handling routine.
- Sample predefined exceptions:
  - **NO\_DATA\_FOUND**
  - **TOO\_MANY\_ROWS**
  - **INVALID\_CURSOR**
  - **ZERO\_DIVIDE**
  - **DUP\_VAL\_ON\_INDEX**

# Predefined Exception

- **Syntax**

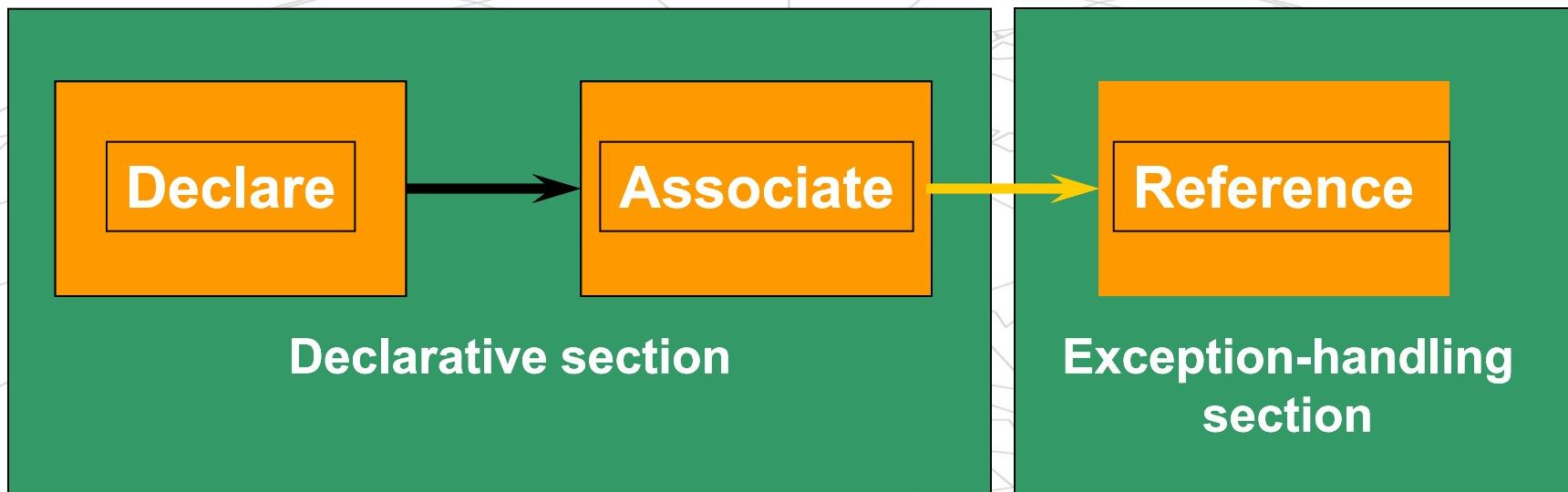
```
BEGIN
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        statement1;
        statement2;

    WHEN TOO_MANY_ROWS THEN
        statement1;

    WHEN OTHERS THEN
        statement1;
        statement2;
        statement3;

END;
```

# Trapping Non-Predefined Oracle Server Errors



- Name the exception
- Code the PRAGMA EXCEPTION\_INIT
- Handle the raised exception

# Non-Predefined Error

- Trap for Oracle Server error number -2292, an integrity constraint violation.

```
DECLARE
```

```
    e_emps_remaining      EXCEPTION;  
  
    PRAGMA EXCEPTION_INIT (  
        e_emps_remaining, -2292);  
  
    v_deptno      dept.deptno%TYPE := &p_deptno;  
  
BEGIN  
    DELETE FROM dept  
    WHERE      deptno = v_deptno;  
    COMMIT;  
  
    EXCEPTION e_emps_remaining  
    WHEN e_emps_remaining THEN  
        DBMS_OUTPUT.PUT_LINE ('Cannot remove dept ' ||  
            TO_CHAR(v_deptno) || '. Employees exist. ');  
  
END;
```

1

2

3

## Summary

- **Exception types:**
  - Predefined Oracle Server error
  - Non-predefined Oracle Server error
- **Exception trapping**
- **Exception handling:**
  - Trap the exception within the PL/SQL block.
  - Propagate the exception.



अॅक्टस  
acts

# Thank You !